

## 01/05/21 16:42

**THIS PAGE BLANK (USPTO)**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-134506

(43) 公開日 平成11年(1999) 5月21日

(51) Int.Cl. <sup>8</sup>	識別記号	F I
G 0 6 T 7/20		G 0 6 F 15/70 4 1 0
G 0 6 M 7/00	3 0 1	G 0 6 M 7/00 3 0 1 B
11/00		11/00 D
G 0 8 G 1/01		G 0 8 G 1/01 A

審査請求 未請求 請求項の数15 O L (全 21 頁)

(21) 出願番号 特願平9-299948

(22) 出願日 平成9年(1997)10月31日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000100997

アキタ電子株式会社

秋田県南秋田郡天王町天王字長沼64

(72) 発明者 雨宮 廣和

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

(72) 発明者 嵯我 俊秀

秋田県南秋田郡天王町字長沼64 アキタ電

子株式会社内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

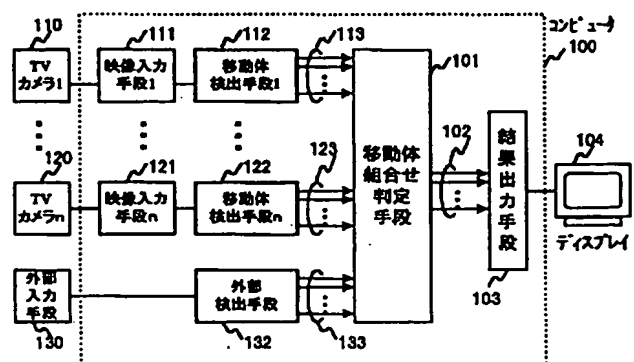
(54) 【発明の名称】 移動体組合せ検出装置および方法

(57) 【要約】

【課題】従来の線状監視方法では、移動体の監視領域通過といった、簡単な条件でしか移動体を検出できなかった。

【解決手段】複数の移動体検出手段を有し、その移動体検出結果の検出位置や検出時間などの組み合わせを元に移動体を検出することにより、より複雑な移動体を検出できる。

図1



## 【特許請求の範囲】

【請求項1】映像を入力する手段と入力された映像から移動体を検出する手段とを組にした手段を少なくとも1つ設け、それぞれの組から出力される移動体検出結果を判定する移動体組合せ判定手段と、検出された結果を出力する手段から構成される移動体組合せ検出装置において、

上記移動体を検出する手段として、入力された映像に対して移動体の有無の判定を行なう着目領域を設定する手段と、ある特定のフレームにおける着目領域のデータと各フレームでの着目領域のデータとの相関を算出する手段と、算出された少なくとも1つの相関値のパターンから、移動体の存在有無や背景画像の変更などの移動体検出イベントを決定する手段を有し、上記移動体組合せ判定手段として、それぞれの移動体検出手段から出力される移動体検出イベントやそれに付属する移動体検出情報を元に最終的な移動体検出を決定する手段を有することを特徴とした移動体組合せ検出装置。

【請求項2】請求項1記載の移動体組合せ検出装置において、移動体組合せ判定手段の移動体検出装置以外の入力として、外部入力手段と外部検出手段を用意し、それを組み合わせて検出することを特徴とする移動体組合せ検出装置。

【請求項3】請求項1記載の移動体組合せ検出装置において、移動体検出手段から出力される移動体検出情報として、複数個ある移動体検出手段を識別する識別子を有することを特徴とする移動体組合せ検出装置。

【請求項4】請求項1記載の移動体組合せ検出装置において、各々の移動体検出装置で用いる着目領域を、直線や曲線といった一筆書きが可能な線状とすることを特徴とする移動体組合せ検出装置。

【請求項5】請求項1記載の移動体組合せ検出装置において、各々の移動体検出装置で用いる着目領域を、枠状や円状といった一筆書きが可能な閉曲線状とすることを特徴とする移動体組合せ検出装置。

【請求項6】請求項1と請求項3に記載の移動体組合せ検出装置において、複数個の移動体検出装置で用いる着目領域を、複数個の線分からなる着目領域として配置することを特徴とする移動体組合せ検出装置。

【請求項7】請求項1と請求項3に記載の移動体組合せ検出装置において、複数個の移動体検出装置で用いる着目領域を、格子状に配置することを特徴とする移動体組合せ検出装置。

【請求項8】請求項1と請求項3に記載の移動体組合せ検出装置において、複数個の移動体検出装置で用いる着目領域を、十字状に配置することを特徴とする移動体組合せ検出装置。

【請求項9】請求項1記載の移動体組合せ検出装置において、映像入力手段と移動体検出手段の組のうち、移動体組合せ判定手段が不必要と判断した組は使用しないこ

とを、移動体組合せ判定装置の判定結果によって、制御できることを特徴とする移動体組合せ検出装置。

【請求項10】請求項1記載の移動体組合せ検出装置において、移動体組合せ判定手段の実現方法として、過去一定時間分の移動体検出情報を格納することを特徴とする移動体組合せ検出装置。

【請求項11】入力された映像に対して移動体の有無の判定を行なう着目領域を設定し、ある特定のフレームにおける着目領域のデータと各フレームでの着目領域のデータとの相関を算出し、算出された少なくとも1つの相関値のパターンから、移動体の存在有無や背景画像の変更などの移動体検出イベントを決定し、上記に示した移動体の検出方法を、それぞれ違う入力映像、着目領域、相関値パターンに対して複数行ない、それぞれの移動体検出方法によって得られた移動体検出有無やそれに付属する移動体検出情報を元に最終的な移動体検出を決定することを特徴とした移動体組合せ検出方法。

【請求項12】指定された映像から移動体の数をカウントする移動体カウント装置において、請求項1から11までに記載の移動体組合せ検出装置と、該装置が出力した検出イベントの数をカウントする装置と、カウントした移動体の数を出力する装置を有することを特徴とした、移動体カウント装置。

【請求項13】請求項1と請求項3に記載の移動体組合せ検出装置において、少なくとも2つの移動体検出装置で得られた移動体検出イベントを用いて、移動体の向きまたは速度を求めることを特徴とする移動体組合せ検出装置。

【請求項14】請求項1と請求項3に記載の移動体組合せ検出装置において、請求項7に記載の格子状の監視着目領域により検出された移動体検出情報の組み合わせから、移動体の位置と移動体の大きさを判定することを特徴とした、移動体位置判定装置。

【請求項15】請求項1記載の移動体組合せ検出装置において、複数の移動体検出装置で用いられる複数個の監視着目領域を1つの画面上で設定することを特徴とした、移動体検出条件設定方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、道路における交通量の計測や、線路・踏み切りにおける犯罪防止など、カメラにより入力された映像監視のための、移動体組合せ検出装置及び方法に関する。

## 【0002】

【従来の技術】現在、道路や踏み切り、銀行のサービスフロント等のいたる場所でカメラ映像による監視が行なわれている。これらは、特定の場所を移動する物体（以下では、移動体または移動体という）を監視することによって、交通集中を解消したり、事故や犯罪を未然に防止することを目的としている。このような移動体に対す

る映像監視のニーズはきわめて高い。ところが、現在の映像監視は技術的な問題から人手に頼らざるをえない状況にある。こうした背景から、コンピュータなどによる監視処理の自動化が求められており、モデルやテンプレート等をを用いた様々な方法が提案されている。

【0003】移動体を検出する方法としては、特開平8-221577号公報に記載される「移動体検出・抽出装置及び方法」が提案されている。この方式では、複雑背景下での移動体検出・抽出と、映像処理時間の短縮とを実現している。以下に図2を用いてこの方式を説明する。

【0004】図2のうち、フレーム画像F1(241)からF5(245)は、それぞれ時刻T1(221)からT5(225)での入力映像のフレーム画像である。図2のフレーム画像中に記されている線分S(231)は、入力映像中での監視を行なう着目領域を線分として指定したものであり、以降、この線分状の着目領域をスリットと呼ぶことにする。図2の201から205は、それぞれ時刻T1(221)からT5(225)までの、スリットS上の画像(以降スリット画像と呼ぶ)と背景画像を表したものである。本例では、処理開始時の背景画像として、移動体がカメラに移っていない時の監視着目領域の画像を設定している。

【0005】本方法では、それぞれのフレーム画像について、次の処理を行なう。(1)ある特定のフレームにおけるスリット画像と背景画像を抽出し、(2)スリット画像と背景画像の画像差分量を、画像のピクセル値の2乗差分などの方法により算出し、(3)該画像差分量を時系列的に眺め、画像差分量が山形のパターン等に従って変移していた場合、移動体と判断し、(4)画像差分量の変動がなく平坦な場合、背景画像と判断する。

【0006】上記のステップ(3)について、図2のフレーム画像列を用いて説明する。本例のように、物体がスリット状を横切る場合、画像差分量は図2の画像変化量グラフ(211)のように山形に遷移する。まず、物体がスリットに入る前(時刻T1(221))では、スリット上の画像と背景画像はほぼ同一(201)なので、画像差分量は小さい。次に、物体がスリットを横切り始める(時刻T2(222))と、スリット画像と背景画像は異なる(202)ため、画像差分量は大きくなる。最後に、物体がスリットを通り抜けると(時刻T3(223))、画像差分量は、また小さい値に戻る。このように物体がスリットを横切ると画像差分量は山形になる。したがって、移動体を見つけるには、画像差分量を時系列的に眺めて、山形になった瞬間を見つければよい。本例では、山形になった瞬間として、画像差分量が、しきい値a(213)を超えて、かつ、その後しきい値a(213)を下回った時点を用いている。

【0007】上記のステップ(4)について、図2のフレーム画像列を用いて説明する。本例のように、スリッ

ト上に荷物(252)などが放置されると(時刻T4(224))、当初画像差分量が増加するが、その後荷物(252)は静止したままなので、画像差分量は、高どまりしたまま変動しなくなる(時刻T4(224)から時刻T5(225))。この方式では、一定時間以上画像差分量の変動値が小さかった場合、その時点のスリット画像を背景としている。

【0008】以上の通り、特開平8-221577号公報では、監視を行なう着目領域として線分を用いることができるため、画面全体を着目領域とするそれ以前の方式に比べ、画像差分量の算出時間を大幅に短縮できる。また、この方式では、画像差分量の時系列変化を見ることにより背景更新のタイミングを見つけることができるため、屋外の映像などのように、背景が常時変更される可能性のある場所でも、監視処理を適用できる。

【0009】

【発明が解決しようとする課題】ところが、上記従来技術を単純に利用する際、次のような問題点が存在する。

【0010】第一の問題点は、監視を行なう着目領域を画面上に1つしか設けられない点である。実際の監視映像では、4台のカメラで別々に撮影した映像を一台のモニタTVで表示できるよう、4映像を2×2の配列状に表示することが多い。この場合、一台のTVモニタ画面上に監視着目領域を4つ指定できることが望ましい。

【0011】第二の問題点は、仮に複数台の移動体検出・抽出装置を用意したとしても、その検出時刻の前後関係や、検出結果画像の類似性の判定など、移動体の内容に基づいた高度な検出判断が行なえないことである。

【0012】

【課題を解決するための手段】まず、本発明を構成する手段として、映像を入力する手段と入力された映像から移動体を検出する手段とを組にした手段を複数個設け、それぞれの組から出力される移動体検出結果を判定する移動体組合せ判定手段と、検出された結果を出力する手段を有する。

【0013】それぞれの移動体検出手段では、移動体の有無や背景更新などを出力するだけでなく、その移動体検出手段の識別子や、検出時刻、検出イベントの種類、および検出判定に用いたスリット画像といった移動体検出情報も出力できるようにする。移動体組合せ判定手段では、それぞれの移動体検出手段から出力された情報から総合的な条件判断などを行なうことによって、最終的な移動体検出を決定する。

【0014】このように、移動体検出情報と移動体組合せ判定手段とを新たに加えることにより、本発明の第一の目的である、入力映像中に複数個の着目領域を設けることを実現でき、さらに本発明の第二の目的である、移動体の検出時刻の前後判定や検出結果画像の類似性の判定など、移動体の内容に基づいた高度な検出判断を行なわせることができる。移動体組合せ判定手段の補助入力

として、赤外線通り抜けセンサーや、カメラ画像が切り替わる映像変化点検出装置など、移動体検出装置以外の装置を設ければ、単独の映像解析方法では難しい複雑な移動物の判定も可能となる。

【0015】その他の特徴的な移動体組合せ検出装置及び方法は、明細書中の記載から自ずと明らかになるであろう。

【0016】

【発明の実施の形態】以下に本発明の実施例について次の4つの例を細かく説明する。4つの実施例として、

(1) 移動体組合せ判定手段の1実施例、(2) 2本のスリットを用いた、移動体の向きと速度の判定方法と、それを用いた移動体カウント装置の1実施例、(3) 複数のスリットを格子状に配置させることによって物体の位置を追尾する追尾監視カメラの1実施例、(4) 複数本のスリットを設定する画面入力方法の1実施例、について述べる。

【0017】本発明の1つ目の実施例として、移動体組合せ判定手段の1実施例について以下に述べる。

【0018】図1は、移動体組合せ検出装置の構成図である。図2は、図1の各々の移動体検出手段の実現方法である。図3は、移動体組合せ判定手段の処理手順の例である。図4は、移動体組合せ判定手段で用いる、移動体検出情報の列の例である。なお以降では、移動体検出情報の列とイベントリストを呼ぶ(移動体検出情報をイベント情報と呼ぶ場合もある)。

【0019】まず図1の移動体組合せ検出装置(100)の構成内容について述べる。移動体組合せ検出装置(100)は、以下の手段から構成される。1つ目の映像入力手段1(111)からn個目の映像入力手段n(121)までの映像入力手段は、各々TVカメラ1(110)からTVカメラn(120)までの複数の映像作成手段によって作成された映像を、移動体組合せ検出装置(100)内に読込む機能を実現している。この場合、TVカメラ1(110)の映像は映像入力手段1(111)にへと入力され、以下同様に、1からnまでの番号iに対して、i番目のTVカメラiの映像は映像入力手段iへと入力される。次に映像入力装置1(111)に読込まれた映像は、映像を構成するフレーム画像の列として、移動体検出手段1(112)に入力され、移動体の有無が検出される。以下同様に、i番目の移動体検出手段iにおいて、映像入力装置iに読込まれた映像は、移動体検出手段iに入力され、移動体の有無が検出される。

【0020】移動体検出手段1(112)は、入力されたある特定のフレームにおける着目領域のデータと各フレームでの着目領域のデータとの相関を算出し、算出された少なくとも1つの相関値のパターンから、移動体の存在有無や背景画像の変更などの移動体検出イベントを判定する。この実現方法として、本実施例では、「従来

の方法」で述べた図2に示される方法を用いるものとする。移動体検出手段1(112)の出力(113)には、移動体がスリットに触れた時点や、移動体がスリットを抜け出した時点や、背景が更新された時点など、相関値のパターンとして求めることが可能な、様々な移動体検出イベントを想定することができる。たとえば図2を用いて説明すると、移動体がスリットに触れた時点は、図2の画像変化量があるしきい値a(213)を超えた時点(時刻T2(222))として検出できる。また同様に、移動体がスリットを抜け出した時点は、画像変化量がしきい値a(213)を超えた後に、また画像変化量がしきい値a(213)を下回った時点(時刻T3(223))として検出できる。また同様に、背景が更新された時点は、画像変化量があるしきい値a(213)を超えた後、数秒間画像変化量が動かなかった時点(時刻T5(225))として検出できる。

【0021】なお、移動体検出手段1(112)以外の移動体検出手段i(iは2からn)についても、同様な方法によって移動体を検出し、その検出結果を出力するものとする。

【0022】移動体検出手段1(112)から移動体検出手段n(122)までのそれぞれの移動体検出手段は、検出した移動体検出イベントを、移動体検出イベントの種類や発生時刻などの移動体検出情報とともに、移動体組合せ判定手段(101)に入力する。本例での移動体検出情報は、図4のイベントリストのうちの1要素(401)に示されるような、移動体検出手段の識別子id(452)(以降ではスリットIDと呼ぶ。文字列などにより実現する)、検出時刻time(453)、スリットに触れた時点や背景更新の時点などの検出した移動体検出イベントの種類type(454)、移動体検出処理に用いたスリット画像slit(455)、同じく背景画像bgr(456)、および、同じくフレーム全体の画像img(457)、から構成されている。

【0023】本例では、スリット画像slit(455)、背景画像bgr(456)、フレーム画像img(457)は、実際の画像データを指すポイントとして実現されており、それぞれ画像411、412、413を指している。

【0024】なお、ここに述べた移動体検出手段以外にも、移動体組合せ判定手段(101)の入力として、赤外線による物体通過センサーや、カメラの切り替わりを自動的に認識できるシーン変化点検出手段など、外部入力手段(130)から入力されたデータを解析する外部検出手段(132)も追加してもよい。

【0025】次に、移動体検出手段などから出力された、移動体検出イベントやそれに付随する移動体検出情報(113、123、133など)は、移動体組合せ判定手段101により、その情報の組み合わせの条件が判定され、判定された結果が、より高度な移動体検出イベ

ント(102)となって出力される。結果出力手段(103)では、より高度な移動体検出イベント(102)を、ディスプレイ装置(104)などによって最終的な移動体組合せ検出結果をユーザに提示する。

【0026】なお、本例では、上記の移動体組合せ検出装置(100)を、メモリと入出力装置と演算装置などから構成されるコンピュータによって実現している。

【0027】次に、本例での移動体組合せ判定手段(101)の実現方法について、図3と図4を用いて細かく説明する。図3の手続き300は、移動体組合せ判定手段(101)に接続された、 $n$ 個の移動体検出手段(112、122など)、または、外部検出手段(132)のうちの、いずれかで移動体検出イベントが発生した際に実行される処理であり、その入力データは、先に述べた移動体検出情報(ここでは変数 $event$ として実現している)である。図4は、移動体組合せ判定手段が内部に持つ移動体検出イベントの列(イベントリスト)である。イベントリストは、先に述べた移動体検出手段が作成して移動体検出情報を、リスト構造により複数個格納している。イベントリストは、その先頭を表す先頭ポインタ400を持ち、そのリスト構造の1要素として1つの移動体検出情報を格納し、各要素間をポインタによって連結している。図4の例では、イベントリストの要素として、要素401と要素402があり、それらは先頭ポインタ400、要素401の次ポインタ領域 $next$ (451)などにより、連鎖的に繋がっている。要素の移動体検出情報の各項目(符号452から符号457まで)は先に述べたのでここでは説明を割愛する。

【0028】図3の手続き300についてステップを追いつながりながら説明する。手続き300は、大きく分けて、過去 $T$ 秒分のイベントを保持する処理と、過去 $T$ 秒分のイベントを組み合わせた条件を判定する処理との2つの部分からなる。

【0029】まず、過去 $T$ 秒分のイベントを保持する処理の部分を説明する。イベントリストの先頭ポインタを用いて、イベントリストの先頭項目にアクセスし、その位置を1イベントを表す変数 $e$ に代入する(301)。次に変数 $e$ を用いて、イベントリストの終わりまで順々に読むループを行なう(302)。なお、イベントリストの最後の要素の次ポインタ領域には、値 $nil$ が入っているものとする。ループ(302)では、次の処理を行なう。

【0030】初めに一時変数 $nx$ にイベントリストの次の要素を待避しておく(311)。次に本処理の実行時間と現在のリスト位置のイベント $e$ の時間差を求めるために、入力イベント $event$ の時間 $event\_time$ と、現リスト位置 $e$ の時間 $e\_time$ の差を求める(312)。求めた時間長があらかじめ決められた時間長 $T$ より大きい場合(314)、その時点のイベントをイベントリストから削除する(321)。ループ30

2の最後の処理として、先ほど待避しておいたリストの次要素位置 $nx$ をイベントの変数 $e$ に再代入し(315)、次のリスト要素に対して同様の処理を繰り返す。ループが終了したら、イベントリストの先頭に入力イベント $event$ を追加する(302)。以下の処理により、イベントリストのうち過去 $T$ 秒より古いイベント情報はリストから削除され、イベントリストの時間長が $T$ 秒以内になる。

【0031】次に、過去 $T$ 秒分のイベントを組み合わせた条件を判定する処理を説明する。ループ304では、あらかじめ用意しておいたイベント組合せ条件の個数の分だけ、次の処理を繰り返し、また何回目の繰り返しであるかを変数 $i$ にセットする(304)。ループの中では、まず、あらかじめ用意しておいたイベント組合せ条件の $i$ 番目の条件の判定処理を実行する(317)。この判定処理(317)は、移動体組合せ検出装置の対象とする移動体の内容により、様々な処理に置き換えることができる。それらの詳しい実現例については後述する。本例では、判定処理317の入力として、先に作成した時間長 $T$ 秒分のイベントリストを与え、判定処理317の出力として、移動体の検出有無を表すフラグと、移動体を検出した場合のその移動体の移動体検出情報 $e\_out$ を得ることにしている。判定処理317の後、判定結果が真ならば(318)、移動体検出イベントを発行し、その移動体検出情報として判定処理317にて得た $e\_out$ を出力する。

【0032】これらの処理をあらかじめ用意したイベント組合せ条件の数だけ繰り返すことにより、1つのイベントリストから、複数の種類の移動体検出イベントを見つけることができるようになる。

【0033】なお、イベントリストの時間長 $T$ と、イベント組合せ条件とを、どのように設定すればよいのかは、以降の実施例において詳しく説明する。

【0034】本発明の2つ目の実施例として、2本のスリットを用いた、移動体の向きと速度の判定方法と、それを用いた移動体カウント装置の1実施例について以下に述べる。

【0035】図5は、移動体組合せ検出装置の入力となる映像を、1つのTVカメラから入力するようにした場合のシステム構成例であり、本実施例以降の実施例での基本システム構成である。図6は、2本のスリットを用いた、移動体の向きと速度の判定方法でのスリット配置方法である。図7は、2本のスリットを用いた、移動体の向きのイベント組合せ条件判定の説明図である。図8は、2本のスリットを用いた、移動体の向きと速度の判定方法の、イベント組合せ条件判定の処理フロー図である。図9は、2本のスリットを用いた、移動体カウント装置の画面出力例である。

【0036】図5は、図1とは別のシステム構成例であり、1つのTVカメラ(501)の映像中に複数のスリ

ットを指定できるようにしたシステム構成である。本構成により、一台のTVカメラ中に現れた移動体を、 $n$ 個の移動体検出手段を用いて、つまり $n$ 本のスリットを用いて、従来の1本のスリットによる検出方法では検出できなかった、移動体の向きや映像中の移動体の場所などの、高度な移動体検出を行なうことができる。本例では、映像入力手段1(511)から映像入力手段 $n$ (521)までの $n$ 個の映像入力手段は、同一のTVカメラ(501)から映像を入力するが、入力された映像に対しては、各々の映像入力手段に対応する $n$ 個の移動体検出手段(512、522など)によって処理される。移動体組合せ判定手段502では、 $n$ 個の移動体検出手段(512、522など)の出力を元に最終的な移動体の検出を判定し、その結果を、結果出力手段503を用いて、ディスプレイ装置504などにより、ユーザに提示する。その他、図5の詳細な実現内容については、図1の説明から自ずと明らかである。

【0037】図6は、2本のスリットを用いた、移動体の向きと速度の判定方法のスリット指定方法である。本例では、車の交通量を調査するために、道路を通行する車に対してTVカメラを向けている。本システムは図5のシステム構成を用いて実現されているものとし、図5の中の移動体検出手段の個数 $n$ は、本例では2となる。図6のTVカメラ映像601には、左向きに走行する車621と、右向きに走行する車622が映っている。本実施例では、移動体の速度とその向きとを高い精度で判定するために、移動体検出手段1が監視するスリットSL(611)と、移動体検出手段2が監視するスリットSR(612)の2本のスリットとを、距離 $L$ メートル(613)を隔てて並べて配置している。

【0038】次に、図7と図6を用いて、図6のようにスリットを配置した際、どのように移動体の向きと速度を判定すればよいのかを、621の左向きの車を例に取しながら、説明する。

【0039】車621がTV映像601の右側から現れ、TV映像601の左側に向かったとすると、まず、まずスリットSR(612)、つまり移動体検出手段2で、移動体検出イベントが発生し、それから少し経った後に、スリットSL(611)、つまり移動体検出手段1で、移動体検出イベントが発生することが分かる。この際に発生する2つの移動体検出イベントに対する移動体検出情報は、スリットSL(611)では移動体検出情報E1(701)のように、スリットSR(612)では移動体検出情報E2(702)のようになる。移動体検出情報には、スリットSR(612)での検出時刻E2.time(722)の値である「3」と、スリットSL(611)での検出時刻E1.time(712)の値である「5」とが記録されている。以上から、移動体が左向きに通過した場合、必ずE1.time(712)の方がE2.time(722)よりも大き

くなることが分かる。また右向きでは、これの逆が成り立つことが分かる。このため、この移動体検出の時間情報から移動体の向きを判定することができる。

【0040】また自ずと、移動体が、スリットSR(612)とスリットSL(611)の間を、時間 $t = E1.time - E2.time$ の時間で通り抜けたことが判定できるので、移動体の速度 $V$ を、あらかじめ測っておいた距離 $L$ (613)を用いて、 $V = L/t$ として求められる。

【0041】以上のように、移動体の向きと速度を2本のスリットにより判定することができた。しかし、この方式の場合、移動体が2本のスリットの中央位置まで来た後に、折り返して反対側に戻ったり、複数の移動体が同時に画面内に侵入したりした場合、正しい移動体判定ができない。これは、2つのスリットを通過した移動体が同一であるなど、移動体の内容まで判定していないことに原因がある。本例では、上記例にさらに条件を加えることによって、より正確な移動体の向きと速度の判定を行なえるようにしている。この実現方法について、以下に述べる。

【0042】まず、先ほどのスリットの配置条件に、次のような平行・直交配置条件を加える。移動体検出手段1の監視するスリットSL(611)と、移動体検出手段2の監視するスリットSR(612)の2本のスリットを、距離 $L$ メートル(613)を隔てて平行に並べ、かつ、移動体621または622の進行方向または道路に対して垂直の位置に配置する。このようにすることにより、車などが通過した際、スリットSL(611)のスリット画像E1.slit(713)と、スリットSR(612)のスリット画像E2.slit(723)とが、ほぼ同一の画像になるため、移動体検出時に2つのスリット画像の類似度を算出することによって、2つのスリットを通過した移動体が同一か否かを判定することができる。

【0043】以上の検出条件をまとめたものが、左向きの車の検出条件式CL(703)であり、「E1.slit(713)とE2.slit(723)がほぼ同一の画像で、かつ、E1.time > E2.time」となる。ただし、ここの条件式CL(703)は、移動体の向きが左向きである条件なので、検出イベント点の識別子の制約条件、「E1.id(711) = "SL"、かつ、E2.id(721) = "SR"」が、加わることとなる。

【0044】次に、図8を用いて、以上に示した検出条件を満たすイベントを判定するための、実際の処理フローを説明する。手続き801は、図3の移動体組合せ判定手段の処理フローにて説明した、イベント組合せ条件判定処理(317)の、本実施例における処理方式を表している。手続き801では、入力は過去 $T$ 秒分のイベントリストであり、出力は移動体有無を表すフラグ $f$ と



イベントの移動体検出情報 $e_o$ である。また、本例の移動体の向きと速度の検出方法では、図3のイベント組合せ条件判定処理(317)を実行するにあたり、イベントリスト時間長 $T$ を5秒に、イベント組合せ条件数を1に設定する(802)。

【0045】はじめに移動体有無フラグを偽に初期化しておく(811)。その後、手続き内一時変数 $e_t$ にイベントリストの先頭のイベントを設定する(812)。次に、手続き内一時変数 $e$ を用いて、イベントリストの先頭から2番目のイベントから、末尾のイベントに向かってリストをたどっていく処理を行なう。このために、まず、変数 $e$ にイベントリストの先頭の次のイベントを設定し(813)、次に $e$ の値が $nil$ になるまで以下の処理を繰り返す(814)。

【0046】なお、以降においては、イベントリストは、先のイベント組合せ条件判定処理(317)により、最新の検出イベントがイベントリストの先頭に置かれ、また、イベントリストを末尾に向かって行くにしたがって時刻が古い検出イベントとなっていくことに注意されたい。

【0047】ループ(814)の中では、初めに最新のイベント $e_t$ と現在のイベントリストの位置でのイベント $e$ とのイベント識別子と比較し(821)、値が違う場合のみステップ831以降を行なう。本例のイベント識別子には“SL”または“SR”のどちらかしか存在しないため、これにより、最新のイベント $e_t$ の時点より以前に反対側のスリットを通り抜けた移動体が存在したか否かを判定できる。ステップ831に処理が移った場合、最新のイベント $e_t$ での移動体と、現リスト位置 $e$ での移動体が同じかどうかを判定するために、2つのイベント時点のスリット画像 $e.slit$ と $e_t.slit$ との画像差分量を求める(831)。この2つ移動体が同一であると、スリット画像がほぼ同一の画像になるので、画像差分量が小さくなる。この差分量があるしきい値よりも小さい場合(832)、右向きまたは左向きの移動体を検出したと見なし、移動体検出フラグ $f$ を真にする(841)。ステップ(841)の後、出力用の移動体検出情報を設定する処理を行なう(842)。

【0048】イベント情報設定処理(842)では、最新のイベント $e_t$ のスリット識別子を見ることにより、右向き移動物体検出なのか、左向き移動物体検出なのかの判定を行なう。移動体が左向きの場合、先に図6のスリットSR(612)の検出イベント $e$ 、次にスリットSL(611)の検出イベント $e_t$ が発生するはずである。したがって、イベント識別子 $e_t.id$ が“SL”の場合(851)、検出されたイベントは左向きであり、結果、出力する移動体検出情報 $e_o$ の識別子には、“左向き”を格納する(861)。また、イベント識別子 $e_t.id$ が“SL”でない場合(851)、移動体検出情報 $e_o$ の識別子には、“右向き”を格納する(8

62)。イベント情報設定処理の最後の処理として、移動体の速度 $e_o.speed$ や検出時刻 $e_o.time$ やフレーム画像 $img$ を最新のイベント情報 $e_t$ を元に設定する(852)。移動体の速度 $e_o.speed$ については、あらかじめ測定しておいたスリット間の距離 $L$ (図6の613)を、最新のイベント $e_t$ の時間と見つけたイベント $e$ の時間との差で割った値を代入すればよい。

【0049】イベント情報設定処理(842)が終了したら、イベントを検出できたとして、そのままループ814を終了する(843)。

【0050】なお、ステップ821にて、イベントの識別子が同じ場合(どちらも“SL”の場合など)や、ステップ832にて、画像差分量がしきい値より大きい場合は、現リスト位置のイベント $e$ は、最新の移動体検出イベント $e_t$ に対応していないと判断し、イベントリストの末尾に向かってループを続ける。

【0051】ループ814がイベントリストの全てイベントを操作したのにも関わらず、 $e_t$ に対応する移動体検出イベントが見つからない場合は、移動体なしと判断し、移動体検出有無フラグ $f$ を偽に設定し、手続き801を終了する。

【0052】図9は、以上に説明した移動体の向きと速度を判定する方法を用いた移動体カウンタ装置の結果出力例方法を示している。ウィンドウ900は、結果を表示する領域であり、コンピュータのOSなどにより表示される。

【0053】ウィンドウ900には、入力映像を表示する領域901と、移動体の数を数える調査を開始させる調査開始ボタン904と、同じく調査を終了させる調査終了ボタン905と、入力映像901中に指定した2本のスリット902と903との距離を表示する領域906(本例ではあらかじめ測定しておいた「5m」の値が表示されている)と、移動体の数と向きと速度と移動体の画像などの一台ずつの調査結果を表示する領域907と、最終的な移動体の数や移動体の平均速度を表示する領域908が存在する。

【0054】入力映像901と2本のスリット902、903の配置の指定は、図6と同様とする。調査開始ボタン904を押下すると、移動体の向きと台数と速度の調査処理が開始し、調査終了ボタン905を押下すると、該調査処理を終了する。

【0055】以下に、該調査処理について簡単に説明する。調査の開始時には、右向きの車両台数と、左向きの車両台数と、合計速度値を0に初期化する。その後移動体を検出するたびに、結果領域907に移動体検出情報を追加表示する。本例では、移動体検出情報の表示方法として、図中の921のように、上から移動体画像、移動体検出時刻、移動体の速度、移動体の向きを順に表示している。

【0056】移動体検出時の処理では、この他にも、右向きの車両台数または左向きの車両台数をカウントする処理と、移動体の平均速度を計算する処理と、これらの処理結果を、合計結果表示領域908に表示する処理とを行う。移動体の平均速度の計算方法については、移動体を検出することに移動体の速度値を合計速度値に累算し、その合計速度値をこれまでの全移動体台数（右向き台数と左向き台数の和）で割ることによって求めればよい。

【0057】本発明の3つ目の実施例として、複数のスリットを格子状に配置させることによって物体の位置を追尾する追尾監視カメラの1実施例について以下に述べる。

【0058】図10は、本追尾監視カメラで用いるスリットの格子状配置方法と、そのスリットを用いた、移動体の位置を判定する条件を説明する図である。図11は、図10で示した移動体位置判定方法を用いた追尾監視カメラの画面表示例である。図12は、追尾監視カメラの移動体イベント組合せ条件を判定する処理フローである。なお、本追尾監視カメラは、図5に示したシステム構成により実現され、また、図5の移動体組合せ判定手段は、図3の全体フローと、図12のフローで示す手続き1201によって実現されるものとする。

【0059】図10は、本追尾監視カメラで用いるスリットの格子状配置方法と、そのスリットを用いた、移動体の位置を判定する条件を説明する図である。本例では、入力されたTV映像1000の中に存在する移動体1041の縦の位置と横の位置を、格子状に配列されたスリット群（1011から1015と、1021から1024）を用いて検出する。

【0060】このスリット群は、縦線の形状のスリットを複数並べた、スリットV1（1011）、スリットV2（1012）、スリットV3（1013）、スリットV4（1014）、スリットV5（1015）から構成される縦スリット群と、同様に横線の形状のスリットを複数並べた、スリットH1（1021）、スリットH2（1022）、スリットH3（1023）、スリットH4（1024）から構成される横スリット群とを、直交に配置することによって、構成される。縦スリット群の各々のスリットは、幅 $L_w$ （1032）の間隔で平行に並んでいる。同様に、横スリット群の各々のスリットは、高さ $L_h$ （1031）の間隔で平行に並んでいる。

【0061】なお、これらのスリットを実現するための、図5のシステム構成中の移動体検出手段1（512）から移動体検出手段n（522）は、スリットに移動体が重なった場合、移動体検出イベントを発行するものとする。具体的には、図2の移動体検出手段の実現方法において、画像変化量がしきい値 $a$ （213）を超えている間、つまり時刻 $T_2$ （222）から時刻 $T_3$ （223）までの区間と、時刻 $T_4$ （224）以降の区間に

おいて、常に移動体検出イベントが発行されるものとする。また、移動体検出情報のスリット識別子として、図10に示したスリットの一本一本を識別するために、“V1”、“V2”、“H1”、“H4”といったスリットの名称に値する文字列を設定されるものとする。

【0062】次に、以上で述べたスリット群を用いて移動体の存在する位置を判定する方法について説明する。スリットV2（1012）とスリットH2（1022）の交点上に移動体が存在する場合、スリットV2（1012）とスリットH2（1022）の両方に移動体検出イベントが発生することとなる。このように、ある番号 $x$ と $y$ において（ $x=1\sim5$ 、 $y=1\sim4$ ）、移動体が存在する場合、スリット“V $x$ ”とスリット“V $y$ ”にて移動体検出イベントが発生することが分かる。ここで、“V $x$ ”という表記は、番号 $x$ の値により、“V1”“V2”“V3”“V4”“V5”と変化するスリット識別子を表し、同様に“H $y$ ”という表記は、“H1”“H2”“H3”“H4”のスリット識別子を表すものとする。以降にて同様の表記をした場合、ここに示した意味を表すものとする。

【0063】以上をまとめると、ある位置（ $x$ 、 $y$ ）での移動体検出条件 $C_{xy}$ （1001）は、ある2つのイベントに対する移動体検出情報 $E_1$ 、 $E_2$ を用いることにより、次のように定義される：「 $E_1.id = \text{“V”}x$ 」かつ $E_2.id = \text{“H”}y$ 」、但し、 $|E_1.time - E_2.time| < \Delta t$ 」。ここで、「 $|E_1.time - E_2.time| < \Delta t$ 」とは、移動体検出イベント $E_1$ と移動体検出イベント $E_2$ が、ほぼ同時期に発生したことを表す制約条件を表し、 $\Delta t$ はあらかじめ固定値を設定しておくものとする。

【0064】図11は、図10で示した移動体位置判定方法を用いた追尾監視カメラの画面表示例である。追尾監視カメラを実現するウィンドウ1101は、入力されたTVカメラ映像を表示する領域1110と、TVカメラ映像のうち、移動体1114が存在する部分1113だけを拡大表示する拡大映像表示領域1120と、移動体の追尾処理を開始させる追尾開始ボタン1131と、移動体追尾処理を終了させる追尾終了ボタン1132、から構成される。なお、TVカメラ映像表示領域1110中に表示されている格子状の線（1111や1112やそれらに平行する直線）は、スリットを表している。これらスリットの詳細な配置方法や実現方法は、図10の説明から自ずと明らかである。

【0065】図12は、図11の追尾監視カメラの実現方法のうち、イベント組合せ条件判定の処理について、詳しく説明するものである。手続き1201は、図3に示した移動体組合せ判定手段の処理フローの中のステップ317から呼ばれる手続きであり、入力は過去 $T$ 秒間のイベントリストであり、出力は移動体の検出有無フラグ $f$ と、移動体検出情報 $e_o$ である。なお、本追尾監視

カメラでは、イベントリスト時間長 $T$ を極短い時間である0.1秒に、イベント組合せ条件の数を1に指定している(1202)。

【0066】手続き1201は、大きく分けて、イベントリストを縦スリット群のイベントリストと横スリットのイベントリストに分類する処理と、その後分類された縦横のイベントリストの組み合わせから、移動体の存在位置を判定していく処理の2つから構成される。

【0067】まず、イベントリストを先頭から末尾にたどりながら、リストの要素 $e$ の識別子が“Hy”である横スリットの検出イベントだけを抽出し、この結果を元に新たなイベントリスト $L_h$ を作成する(1211)。同様に、イベントリストから、リストの要素 $e$ の識別子が“Vx”である縦スリットの検出イベントだけを抽出し、この結果を元に新たなイベントリスト $L_v$ を作成する(1212)。

【0068】その後、以下に示すステップによって、分類された縦横のイベントリストの組み合わせから、移動体の存在位置を判定していく処理を行なっていく。一般に、移動体の存在する縦横スリットの交点(例えば図10のスリットV2(1012)とスリットH2(1022)の交点(1051)など)は、複数存在する。以降の処理では、これら複数個のスリット交点を含む最小の矩形領域を求め、その値を、変数 $x_1$ (矩形左位置を意味する)、 $y_1$ (同矩形上位置)、 $x_2$ (同矩形右位置)、 $y_2$ (同矩形下位置)に代入する処理を行なっている。

【0069】ステップ1214では、まず矩形領域を表す変数 $x_1$ 、 $y_1$ 、 $x_2$ 、 $y_2$ と、スリット交点の個数 $n$ とを初期化する(1214)。以降のステップにて、最小の矩形領域を求めるため、 $x_1$ を $\infty$ 、 $y_1$ を $\infty$ 、 $x_2$ を0、 $y_2$ を0に初期化している。また交点の個数 $n$ を0に初期化している。

【0070】次に、一時変数 $eh$ に横イベントリスト $L_h$ の先頭を代入し(1215)、横イベントリスト $L_h$ の末尾まで読み出していくループ(1216)を行なう。横イベントリストも最後の要素のポインタ値は $nil$ であるので、 $eh$ の値が $nil$ になるまでループする。

【0071】横イベントリストのループ(1216)の中では、まず、移動体イベントの検出情報 $eh$ の識別子 $id$ (この値は先に分類したので必ず“Hy”である)から、横スリットの行番号 $y$ を求め、その行番号 $y$ からスリット“Hy”の $y$ 座標を求め、変数 $s_y$ に代入する(1221)。スリット“Hy”から $y$ 座標を求めるには、例えば、各スリットの識別子とその $x, y$ 座標をテーブル化しておき、イベント情報 $eh$ から得たスリットの行番号または $eh$ の識別子から、テーブルを引くことによって、そのスリットの $x, y$ 座標を求めることができる。

【0072】ステップ1222では、横イベントリストの順次読み出しのために、イベント情報 $eh$ の次ポインタ $eh.next$ を $eh$ に代入している(1222)。

【0073】次のステップ1223、1224によって縦イベントリスト $L_v$ の全要素に対してループを行なっている。まず、縦イベントリスト $L_v$ の先頭要素を変数 $ev$ に代入し(1223)、その $ev$ の値が $nil$ になるまで、縦イベントリストを末尾に向かって読み出すループを行なう(1224)。

【0074】縦イベントリストの読み出しループの中では、縦スリットと横スリットの交点が見つかったとして矩形計算処理を行なう。まず、交点の数を表す変数 $n$ に1を加算する(1241)。次に、移動体イベントの移動体検出情報 $ev$ の識別子 $id$ (この値は先に分類したので必ず“Vx”である)から、縦スリットの行番号 $x$ を求め、その行番号 $x$ からスリット“Vx”の $x$ 座標を求め、変数 $s_x$ に代入する(1242)。このステップの実現方法についても、ステップ1221と同様の手法を適用すればよい。

【0075】ステップ1243では、縦イベントリストの順次読み出しのために、イベント情報 $ev$ の次ポインタ $ev.next$ を $ev$ に代入している(1243)。以降のステップでは、ステップ1221と1242にて求めたスリット交点の座標 $s_x$ 、 $s_y$ を元に、移動体の存在する最小矩形を更新する処理を行なう。

【0076】矩形左位置の更新処理として、スリット交点位置 $s_x$ が、現左位置 $x_1$ より小さい場合(1244)、現左位置 $x_1$ をスリット交点位置 $s_x$ の値に更新する(1254)。

【0077】矩形上位置の更新処理として、スリット交点位置 $s_y$ が、現上位置 $y_1$ より小さい場合(1245)、現上位置 $y_1$ をスリット交点位置 $s_y$ の値に更新する(1255)。

【0078】矩形右位置の更新処理として、スリット交点位置 $s_x$ が、現右位置 $x_2$ より大きい場合(1246)、現右位置 $x_2$ をスリット交点位置 $s_x$ の値に更新する(1256)。

【0079】矩形下位置の更新処理として、スリット交点位置 $s_y$ が、現下位置 $y_2$ より大きい場合(1247)、現下位置 $y_2$ をスリット交点位置 $s_y$ の値に更新する(1257)。

【0080】以上の2つのループ1216と1224を行なうことにより、縦横スリットの交点の数 $n$ と、移動体が存在する最小矩形領域 $x_1$ 、 $y_1$ 、 $x_2$ 、 $y_2$ が求まる。

【0081】本手続き1201の最後の処理として、移動体の存在有無を判定する。縦横スリットの交点の数 $n$ の値が0より大きい場合(1217)、移動体ありと判断し、移動体検出有無フラグ $f$ を真にする(1231)。次に先に求めた最小矩形を元に追尾カメラで映像

を拡大する場所を計算し、その結果を出力する移動体検出イベントの移動体検出情報e oに設定する(1232)。映像を拡大する領域としては、先に求めた最小矩形の上下それぞれにスリットの縦間隔である図10のLh(1031)の半分の余裕領域を追加し、同様に最小矩形の左右それぞれにスリットの横間隔である図10のLw(1032)の半分の余裕領域を追加する。

【0082】縦横スリットの交点の数nの値が0の場合(1217)、移動体なしと判断し、移動体検出有無フラグfを偽にする(1233)。

【0083】以上のようなイベント組合せ条件判定処理を用意することにより、映像中に移動体が存在する場合、図5の移動体検出組合せ判定手段502が移動体検出イベントを発行する。結果出力手段503では、移動体検出イベントが発生した場合、その移動体検出情報に格納されている映像拡大領域に従って、入力となるTVカメラ映像の一部を図11の拡大映像領域1120のように拡大表示する。

【0084】もちろん上記以外にも、あらかじめ用意しておいたもう一台の高精細TVカメラまたは高精細デジタルスチルカメラなどにより、指定された拡大領域の部分を別途詳細に撮影することも可能である。

【0085】さらには、移動体の矩形領域1113の中央xy座標とTVカメラ映像1120の中央xy座標との差分ベクトルから、入力TVカメラの向きやズーム度合いを制御することも可能である。ただしこの場合、TVカメラを動かすと元となる背景画像も更新されるため、TVカメラを動かした際は、一度追尾処理を終了し、再度追尾処理を開始するなどにより、背景を新しく更新する必要がある。なお、該差分ベクトルを求める際の移動体の矩形領域の中央xy座標の代わりとして、スリット交点群の重心のxy座標を用いることができる。重心のxy座標を求めるには、図12のループ1224の際に、スリット交点座標(sx, sy)の累算を行ない、ループ終了後、累算したxy座標をスリット交点数nで割ればよい。

【0086】本発明の4つ目の実施例として、複数本のスリットを設定する画面入力方法の1実施例について以下に述べる。

【0087】図13は、複数本のスリットの条件を設定する画面の1実施例である。図14は、本実施例で用いるスリット位置情報の配列構造である。図15は、本実施例での画面初期化処理を表すフローである。図16は、本実施例での画面表示時のユーザ操作イベントに対応する処理フローである。図17は、複数スリットの条件設定において、入力映像を複数にした場合の設定画面の1実現例である。

【0088】図13は、複数本のスリットの条件を設定する画面の1実施例である。スリット条件設定画面(1300)には、現在選択するスリットの番号を指定する

チェックボックス1(1301)、チェックボックス2(1302)、チェックボックス3(1303)、と、スリットの座標位置を入力可能なエディットボックス群1305と、入力映像とスリット位置を表示する領域(1310)と、スリットの組合せ条件を指定するエディットボックス(1306)と、設定の反映を表すOKボタン(1320)と、設定を取り消すキャンセルボタン(1321)から構成されている。入力映像表示領域(1310)には、現在指定されているスリット群(1311、1312)が表示され、そのうち選択中のスリット(1312)は太線などにより強調表示される。スリット番号を指定するチェックボックス1から3まで(1301、1302、1303)は、どれか1つだけが選択可能になるように設定されている。

【0089】本画面では、3つのスリットの条件を設定できるようになっている。次にその画面の操作方法について概説する。まず、チェックボックス(1301、1302、1303)を指定することにより、現在設定するスリットを1つ選択する。この際、エディットボックス群1305に、各々のスリットの左(変数x1)、上(同y1)、右(同x2)、下(同y2)の現在の座標値が表示されるので、ユーザは必要に応じて数字を修正する。ユーザは必要ならば、チェックボックス(1301、1302、1303)の選択番号を変更して、次のスリット情報を変更できる。複数のスリットの組合せ条件を設定するには、エディットボックス(1306)に、スリット番号「1」「2」「3」と「and」「or」などの演算子を用いた条件式を記述する。図13のスリット組みあわせ条件では、「1 and 2 or 3」となっており、これは、「スリット1とスリット2とで移動体検出イベントが発生し、かつ、スリット3では移動体検出イベントが発生していない場合」、を表している。

【0090】図14は、本実施例で用いるスリット位置情報の配列構造である。スリット位置情報の配列slitpos[] (1401)は、配列の1要素が1本のスリットの位置情報を表している。配列(1401)の1要素には、スリットの左位置(1421、要素x1)、上位置(1422、同y1)、右位置(1423、同x2)、下位置(1424、同y2)が格納されている。図14の配列1401には、スリット1の位置情報slitpos[1] (1411)と、スリット2の位置情報slitpos[2] (1412)と、スリット3の位置情報slitpos[3] (1413)とが格納されている。

【0091】図15と図16は、本実施例でのスリット位置入力方法の処理フローである。図15は、本実施例での画面初期表示処理を表すフローで、図16は、本実施例での画面表示中のユーザ操作イベントに対応する処理フローである。

【0092】まず、図15の画面初期表示処理フローを説明する。ユーザがスリット条件を設定しようとして、図13の画面1300を表示した場合、初期表示処理（1501）実行される。まずループ（1511）により、現在移動体組合せ検出装置で設定されているスリット位置情報、つまり複数個の移動体検出手段の設定情報を、配列 `s l i t p o s` に取得する。ループ（1511、ループカウンタ `i`）内では、`i` 番目の移動体検出手段のスリット位置情報を、スリット位置配列 `s l i t p o s [ i ]` の `x 1`（図14の1421の欄）、`y 1`（同1422）、`x 2`（同1423）、`y 2`（同1424）に設定する。次に、移動体組合せ判定装置から現在設定されている検出条件の文字列を得て、エディットボックス1306に設定する（1512）。この検出条件の取得方法については後述する。次に、入力映像表示領域1310に入力TV映像を常時表示させる（1513）。その後現選択スリット番号の初期化として、選択スリット番号 `s e l` に1を設定し、チェックボタン1301の選択状態をONにする（1514）。その後、現在のスリット位置を表示するために、先に設定したスリット位置情報の配列 `s l i t p o s` と、選択スリット番号 `s e l` とをパラメタとしてスリット表示処理1502を呼び出す（1515）。初期表示処理の最後として、OKボタン（1320）またはキャンセルボタン（1321）が押されるまで、ユーザの画面操作に対応する処理を繰り返す。このために、ループ終了フラグ `f` を用意し、ループ開始前に「偽」に初期化し（1516）、その後ループ終了フラグ `f` が真になるまでループする（1517）。このループ終了フラグ `f` は、OKボタン（1320）またはキャンセルボタン（1321）が押されると「真」になる。ループ内では、キーボードやマウスなどのユーザ操作イベントを取得（1523）した後、ユーザ操作イベントに対応する処理を行う（1524）。なお、スリット位置の表示処理（1502）では、スリット3本分のループ（1531、ループカウンタ `i`）により、3本のスリットを表示する処理をする。ループ1531内では、まずこれから表示するスリット番号 `i` と現在選択しているスリット番号 `s e l` とを比較し（1532）、これから表示するスリットが現在選択するスリットの場合、これから描画する線の太さを太くし（1541）、そうでない場合は、これから描画する線の太さを細くする（1543）。描画する線の太さの設定には、たとえばOSの描画属性の変更によって実現できる。この描画する線の太さの変更の後、`i` 番目のスリット位置情報 `s l i t p o s [ i ]` の値に従って、入力TV映像表示領域（1310）上の座標（`x 1`, `y 1`）から座標（`x 2`, `y 2`）までの線を描画する。

【0093】図16は、本実施例での画面表示時のユーザ操作イベントに対応する処理フローである。ユーザ操作イベント処理（1601）では、まずユーザイベント

の種別を判断して各々の操作イベントに対応する処理を行う（1611）。設定するスリット番号を指定するチェックボタン（1301、1302、1303）の選択状態が変化した場合、選択スリット番号 `s e l` を現在のチェックボックスの番号の値に更新し（1621）、スリットを再描画する（1622）。スリット位置指定エディットボックス群（1305）の値が変化した場合、該エディットボックスの値 `x 1`, `y 1`, `x 2`, `y 2` を、スリット位置配列 `s l i t p o s` の `s e l` 番目である `s l i t p o s [ s e l ]` に格納し（1631）、その後スリット位置の再描画を行う（1632）。OKボタン（1320）が押下された場合、3本分のスリットに対してループ（1641、ループカウンタ `i`）することによって、スリット位置配列の `i` 番目である `s l i t p o s [ i ]` の位置情報 `x 1`, `y 1`, `x 2`, `y 2` を、`i` 番目の移動体検出手段のスリット位置情報として設定する（1661）。その後検出条件を設定するエディットボックス1306の文字列を、移動体組合せ判定装置の検索条件として設定する（1642）。

【0094】ここで文字列を検索条件式として移動体組合せ判定装置に設定する実現方法としては、コンパイラなどで用いる構文解析処理によって、条件文字列を条件式を表した木構造データに変換して、その木構造データを移動体組合せ判定手段に設定すればよい。移動体組合せ判定手段では、入力となる移動体検出手段が出力した移動体検出情報を元に、上記木構造条件式を葉ノードから計算順次根のノードに向かって `and`, `or` などの演算をすれば、最終的な条件の評価値（真または偽）を得ることができる。また逆に、木構造条件式を文字列に変換するには、木構造のツリーワークなど、一般的に知られている方法に従って木構造を文字列化すればよい。

【0095】移動体検出手段のスリット位置情報と、移動体組合せ判定手段の検出条件とを更新した後、ループ終了フラグ `f` を真にする（1643）ことによって、ユーザ操作イベント処理ループ（1516）を終了させる。キャンセルボタン（1321）が押下された場合、移動体検出手段のスリット位置情報の更新は行わず、ループ終了フラグ `f` を真にして（1651）、ユーザ操作イベント処理ループ（1516）を終了させる。

【0096】以上が図13の複数スリット条件設定画面の1実施例であるが、上記方法以外にも次のような実施方法によって、複数スリット条件設定画面を実現することも可能である。たとえば、スリット位置指定エディットボックス群1305によるスリットの位置座標の指定方法を変更して、入力映像表示領域1310上でマウスをドラッグするなどにより、直接的にスリット線の位置を指定することもできる。この場合、ドラッグの開始点とドラッグの終了点をスリットの左右上下の位置とすればよい。

【0097】次の例として、入力映像表示領域1310

での表示方法を上記のようなTV映像ではなく、本設定画面を表示した時点のTV映像中の1フレーム画像を表示することがあげられる。映像表示を静止画表示にすれば、コンピュータの処理負荷を軽減できる。

【0098】その他の例として、移動体組合せ条件設定エディットボックスに指定する条件文に、「1 after 2」などの時間制約条件を指定可能にすることである。この条件文は、「スリット2で移動体検出イベントを発行された後に、スリット1で移動体検出イベントが発生した」ことを表す。これを実現するには、2つ目の実施例「2本のスリットを用いた移動体の向きと速度の判定方法」で示したような、過去のイベントリストをたどって、対応するスリット識別子を探す処理によって実現することができる。

【0099】最後の例として、図17に示されるような、入力TV映像が、1つではなく、複数ある場合の複数スリット条件設定画面があげられる。このような複数の入力映像のスリット条件設定が必要となる場合として、TV会議システムや集中監視センタなどでのTV映像があげられる。図17は、TV会議映像に対するスリット条件設定を行う画面の一部であり、ここでは、図13の入力映像表示領域1310の部分を変更し、多地点映像表示領域1710としている。多地点映像表示領域1710には、4地点のTV会議の映像(1711, 1712, 1713, 1714)が表示され、それぞれの映像に対して、4本のスリット(1731, 1732, 1733, 1734)位置を表示している。

【0100】なお、図17には明記していないが、図17のようなTV会議映像でのスリット設定として、次のような全員着席したことを表す条件を考えることができる。まず、1つの入力映像1711に人物が着席した条件として、「スリット1731に人物が侵入したイベントが発生し、かつ、その後スリット1731にて人物が静止したことによる背景更新のイベントが発生した場合」を定義する。したがって、全員着席の条件は、前記と同様の条件が、4つの入力映像(1711, 1712, 1713, 1714)に含まれる4つのスリット(1731, 1732, 1733, 1734)の、全てにおいて発生した場合として、定義できる。

【0101】

【発明の効果】本発明によれば、入力映像中に複数個の監視着目領域を設けることができるため、1モニタ画像中に複数のTVカメラ画像などを含む入力映像などに対して、どのTVカメラで移動体を検出したのか等を容易に判定できる。

【0102】また、複数個ある監視着目領域を、格子状や平行線上など様々に組み合わせた検出条件を加えることによって、移動体の向きや速度を判断したり、移動体を追尾したり、移動体の静止状態を把握したりするなど、移動体の内容をより精密に判定できるようになる。

【図面の簡単な説明】

【図1】本発明の一実施例における移動体組合せ検出装置の構成図である。

【図2】移動体検出手段の一実現方法を説明する図である。

【図3】本発明の一実施例における移動体組合せ判定手段の処理手順を説明する処理フロー図である。

【図4】移動体組み合わせ判定手段が内部に持つ移動体検出イベントの列(イベントリスト)の一実現例を説明する説明図である。

【図5】本発明の第2の実施例における1台のTVカメラを用いた移動体組合せ検出装置のシステム構成図である。

【図6】本発明の一実施例における2スリットによる移動体の向きと速度の判定方法を説明する図である。

【図7】本発明の一実施例における2スリットによる移動体の向きのイベント組合せ条件判定を説明する説明図である。

【図8】本発明の一実施例における2スリットによるイベント組合せ条件判定の処理フロー図である。

【図9】本発明の一実施例における2スリットによる移動体カウント装置の画面出力例を説明する図である。

【図10】本発明の一実施例における追尾監視カメラで用いる格子状スリット配置方法と移動体の位置の判定方法を説明する図である。

【図11】本発明の実施例における移動体の位置判定方法を用いた追尾監視カメラの画面の表示例を説明する図である。

【図12】本発明の実施例における追尾監視カメラの移動体イベント組合せ判定の処理フロー図である。

【図13】本発明の一実施例における複数本のスリット条件を設定する画面を説明する説明図である。

【図14】本発明の一実施例におけるスリット条件指定画面でのスリット位置情報の配列構造を説明する説明図である。

【図15】本発明の一実施例におけるスリット条件指定画面での画面初期化処理フロー図である。

【図16】本発明の一実施例におけるスリット条件指定画面でのユーザ操作イベントに対応する処理フロー図である。

【図17】本発明の一実施例における複数の映像が入力された場合のスリット条件指定画面を説明する図である。

【符号の説明】

100 コンピュータで実現された移動体組合せ検出装置

101 移動体組合せ判定手段

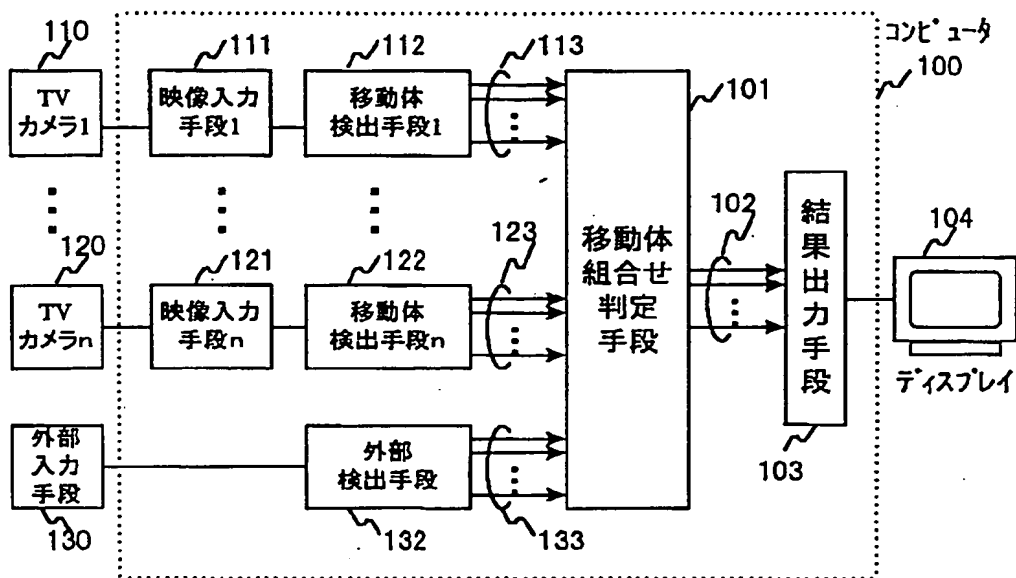
102 移動体組合せ判定手段が出力する移動体検出情報

103 結果出力手段

- 104 ディスプレイ  
 110 1つ目のTVカメラ  
 111 1つ目の映像入力手段  
 112 1つ目の移動体検出手段  
 113 1つ目の移動体検出手段が出力する移動体検出  
 情報  
 110 n個目のTVカメラ  
 111 n個目の映像入力手段  
 112 n個目の移動体検出手段  
 113 n個目の移動体検出手段の移動体検出情報  
 130 外部入力手段  
 132 外部検出手段  
 133 外部検出手段が出力する検出情報

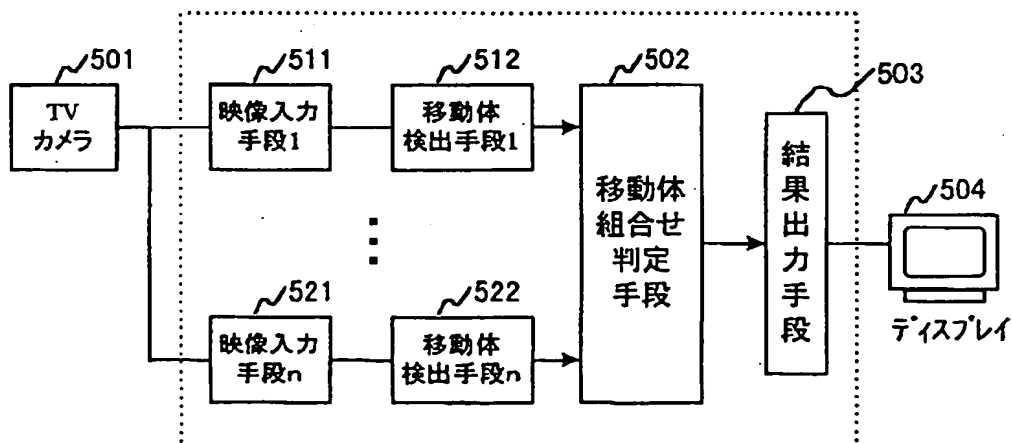
【図1】

図1



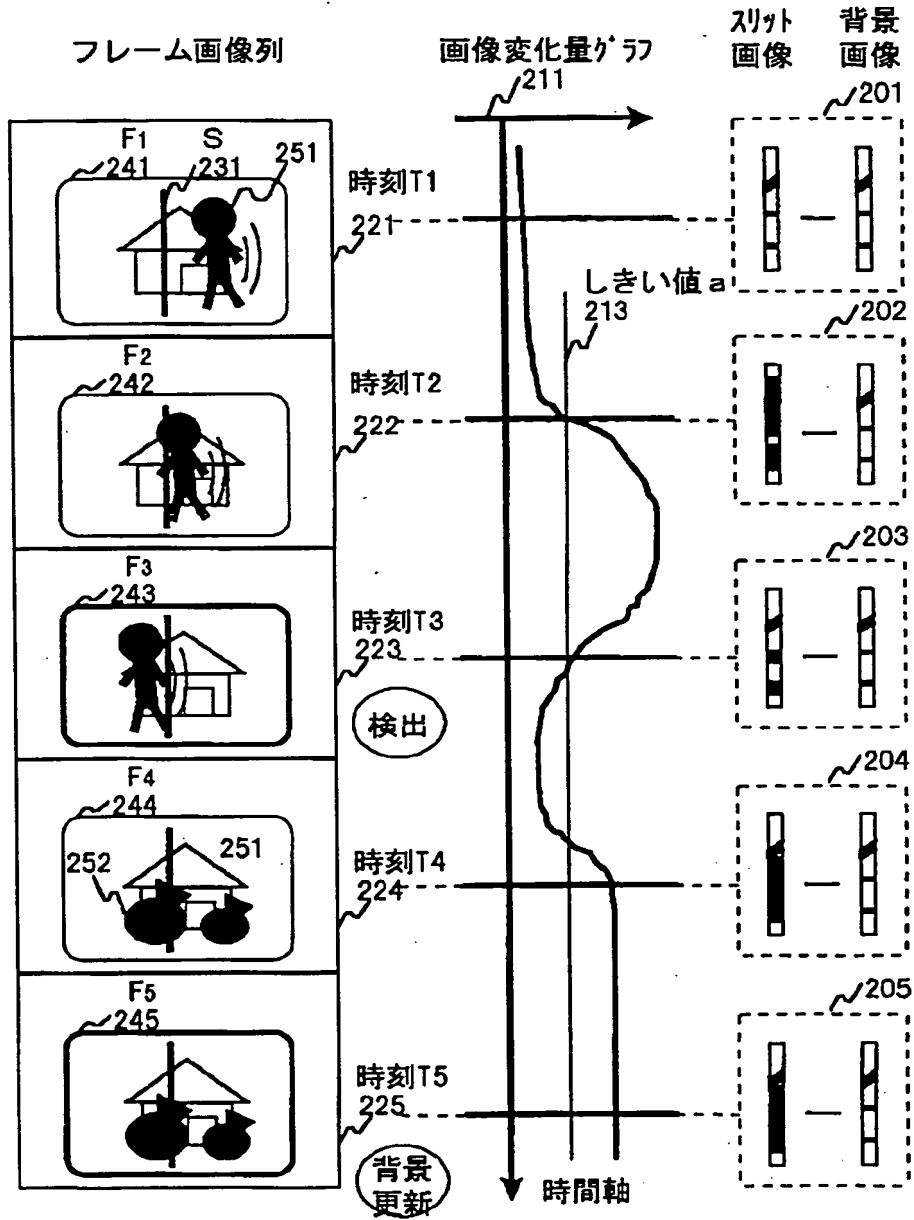
【図5】

図5



【図2】

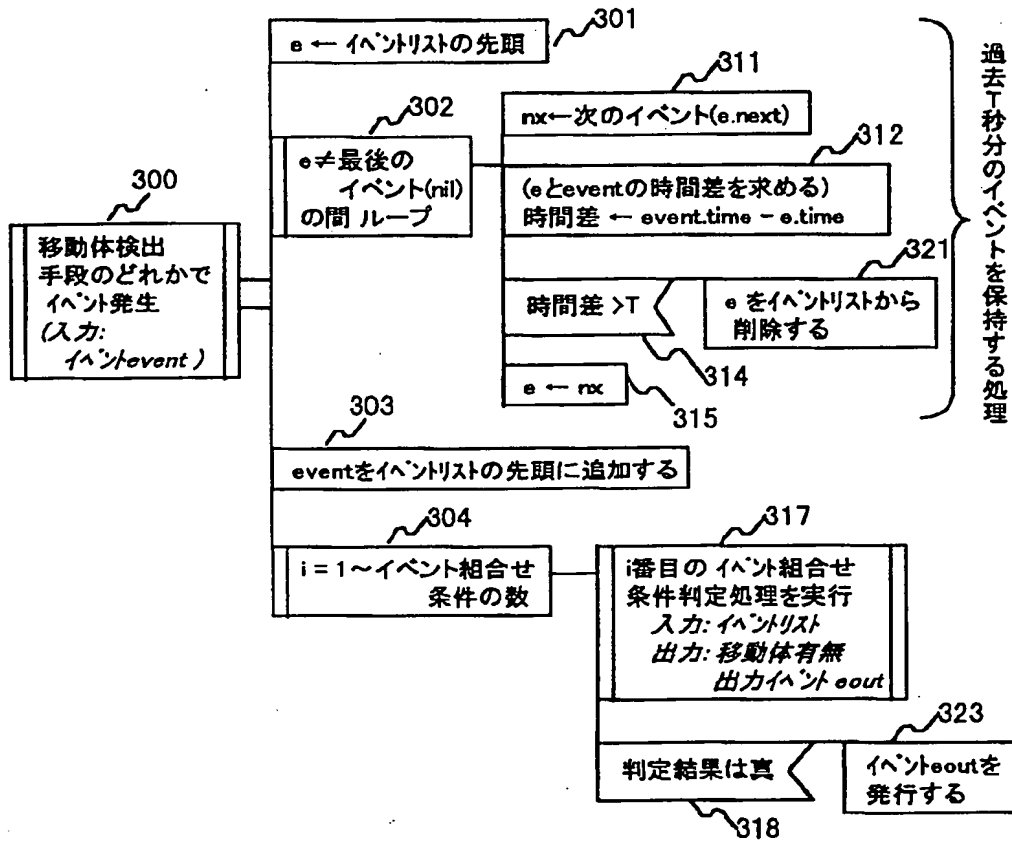
図2





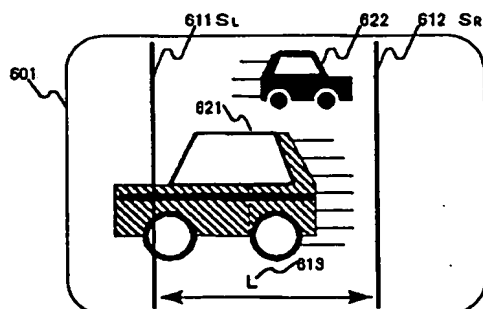
【図3】

図3



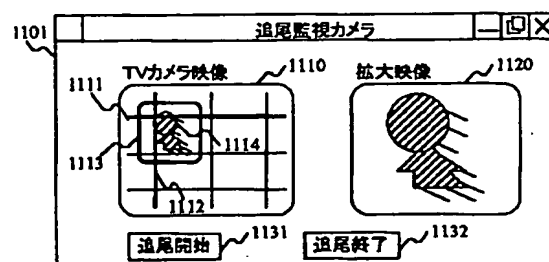
【図6】

図6



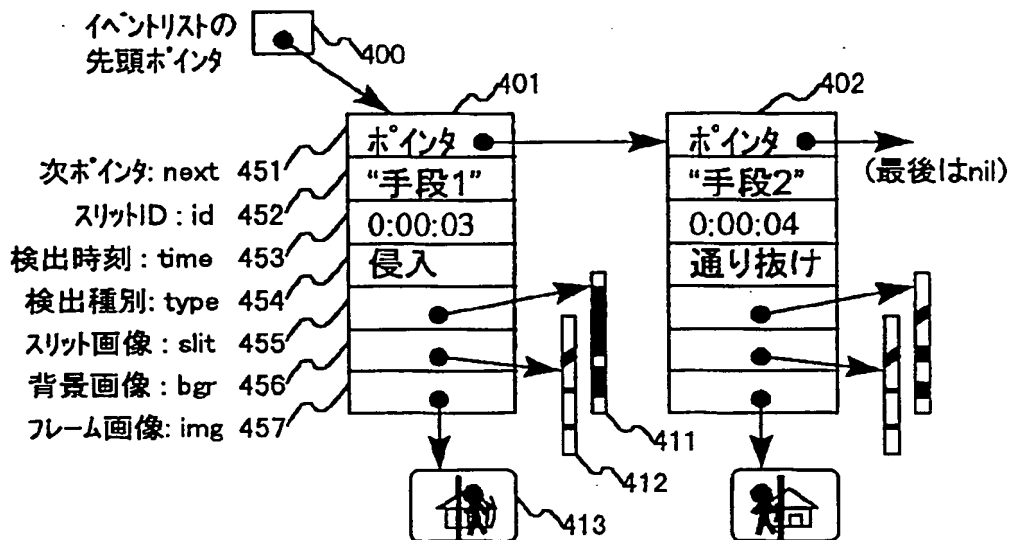
【図11】

図11



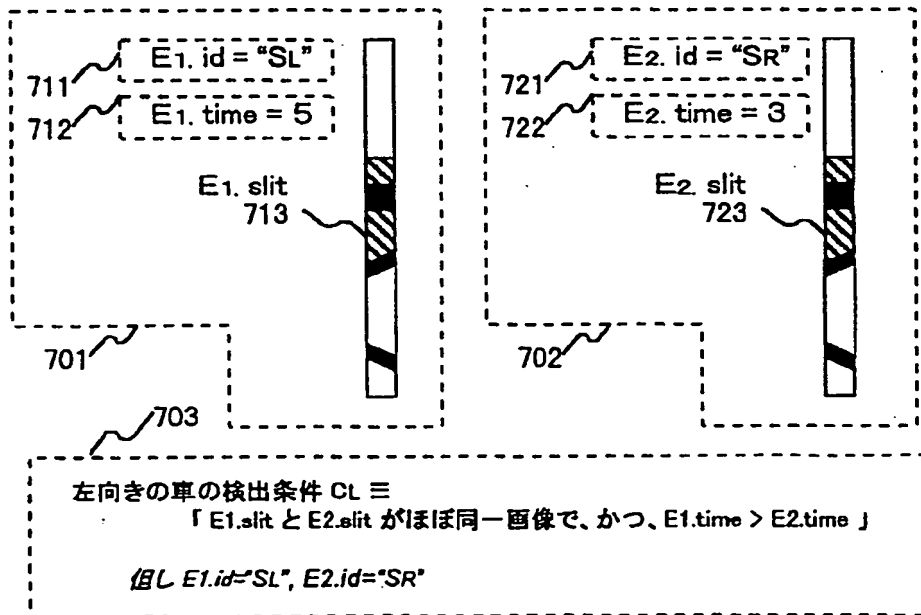
【図4】

図4



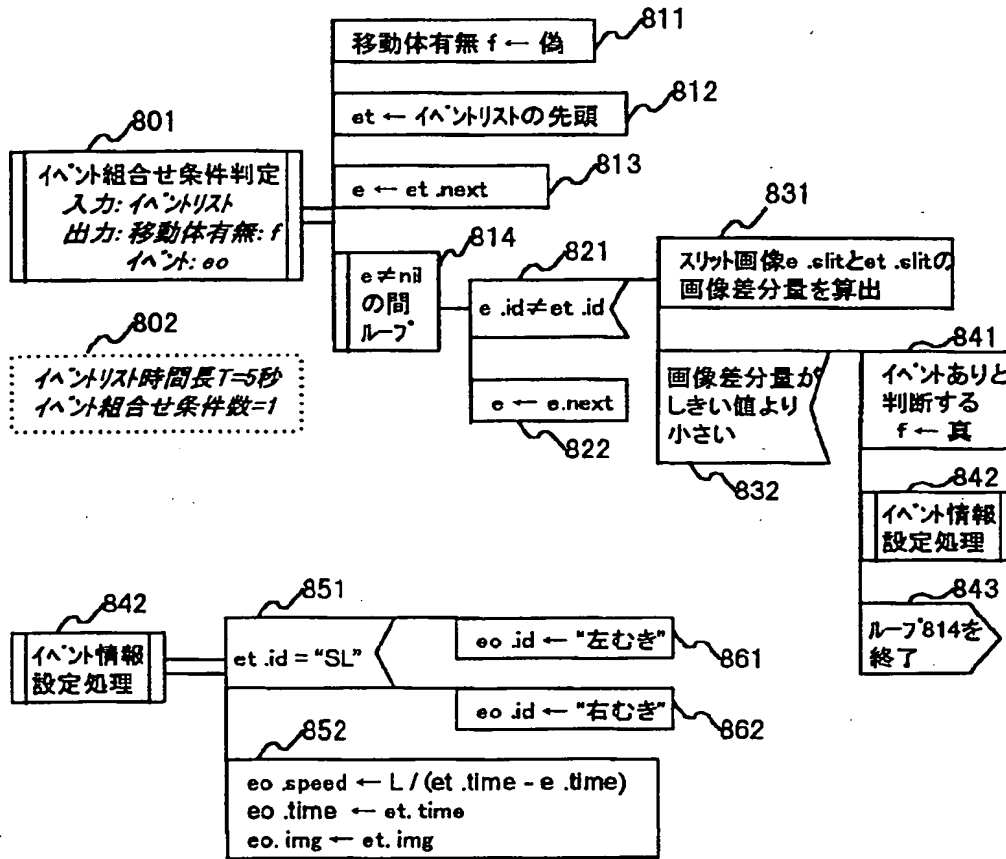
【図7】

図7



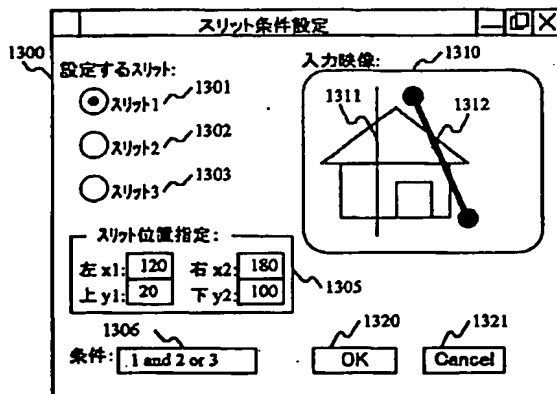
【図8】

図8



【図13】

図13



【図14】

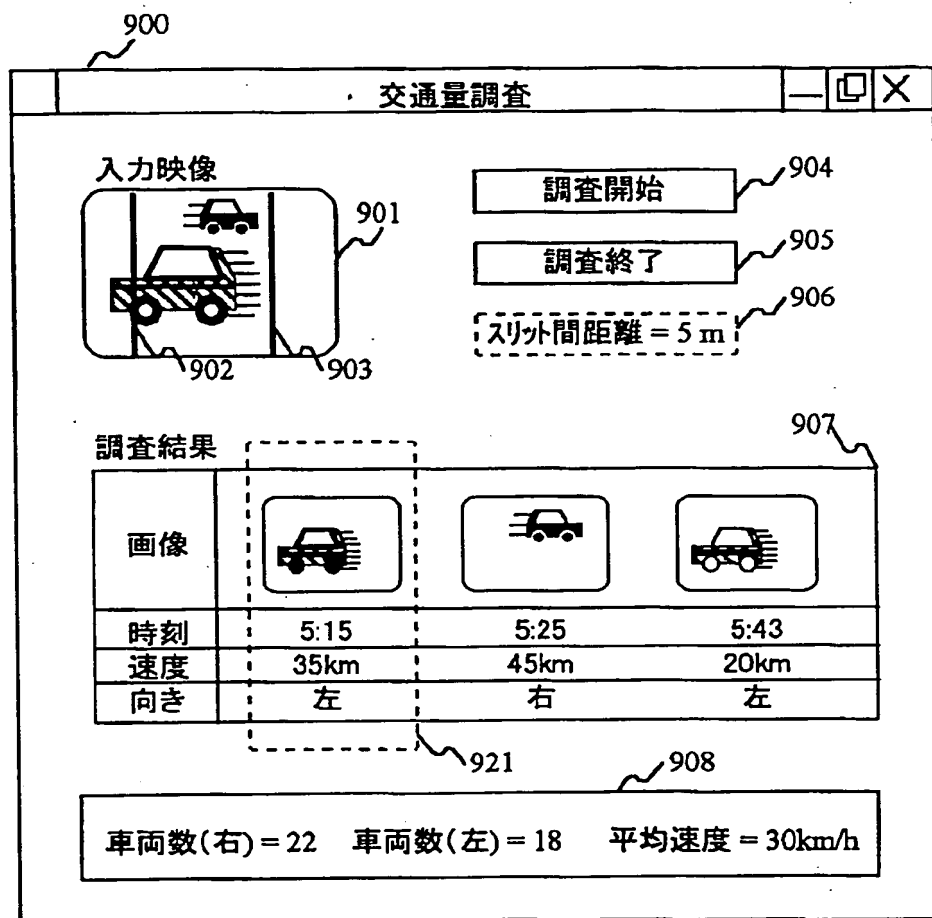
図14

	左 x1 1421	上 y1 1422	右 x2 1423	下 y2 1424
slitpos[1] 1411	70	10	70	110
slitpos[2] 1412	130	10	80	100
slitpos[3] 1413	0	0	160	120

1401 配列 slitpos[ ]

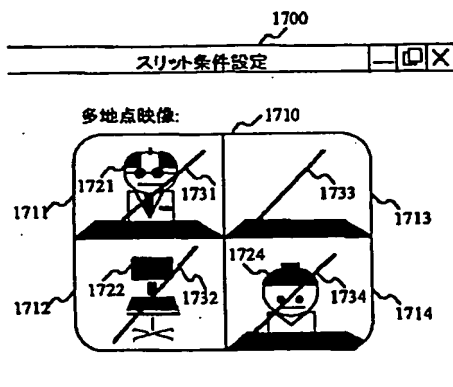
【図9】

図9



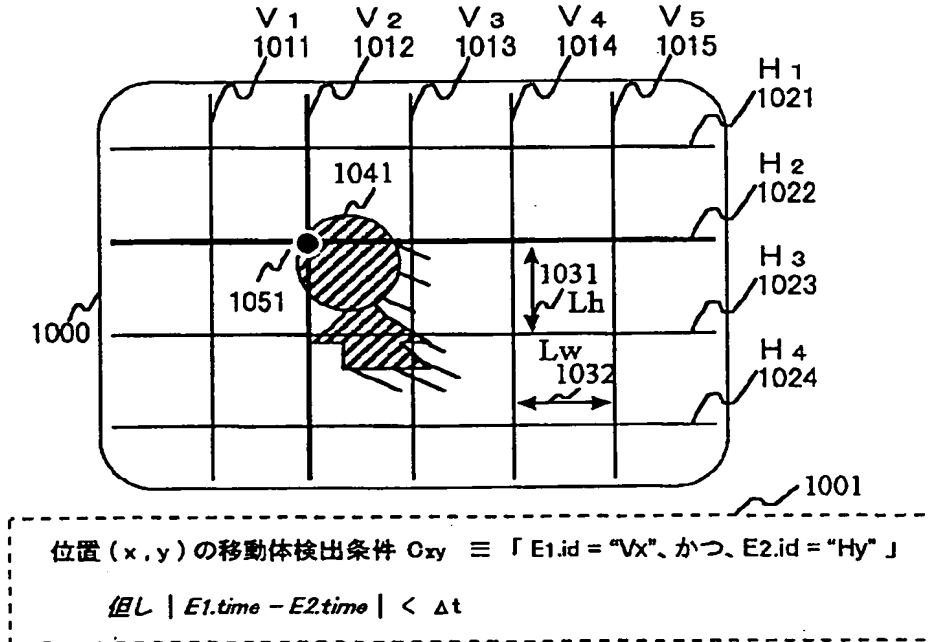
【図17】

図17



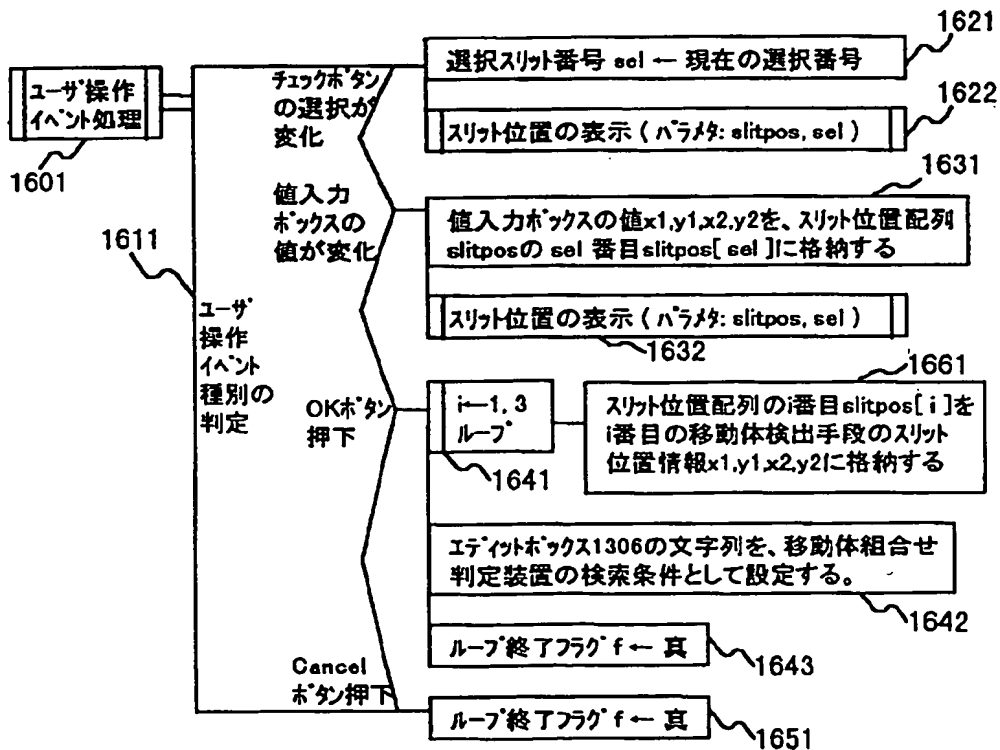
【図 10】

図 10



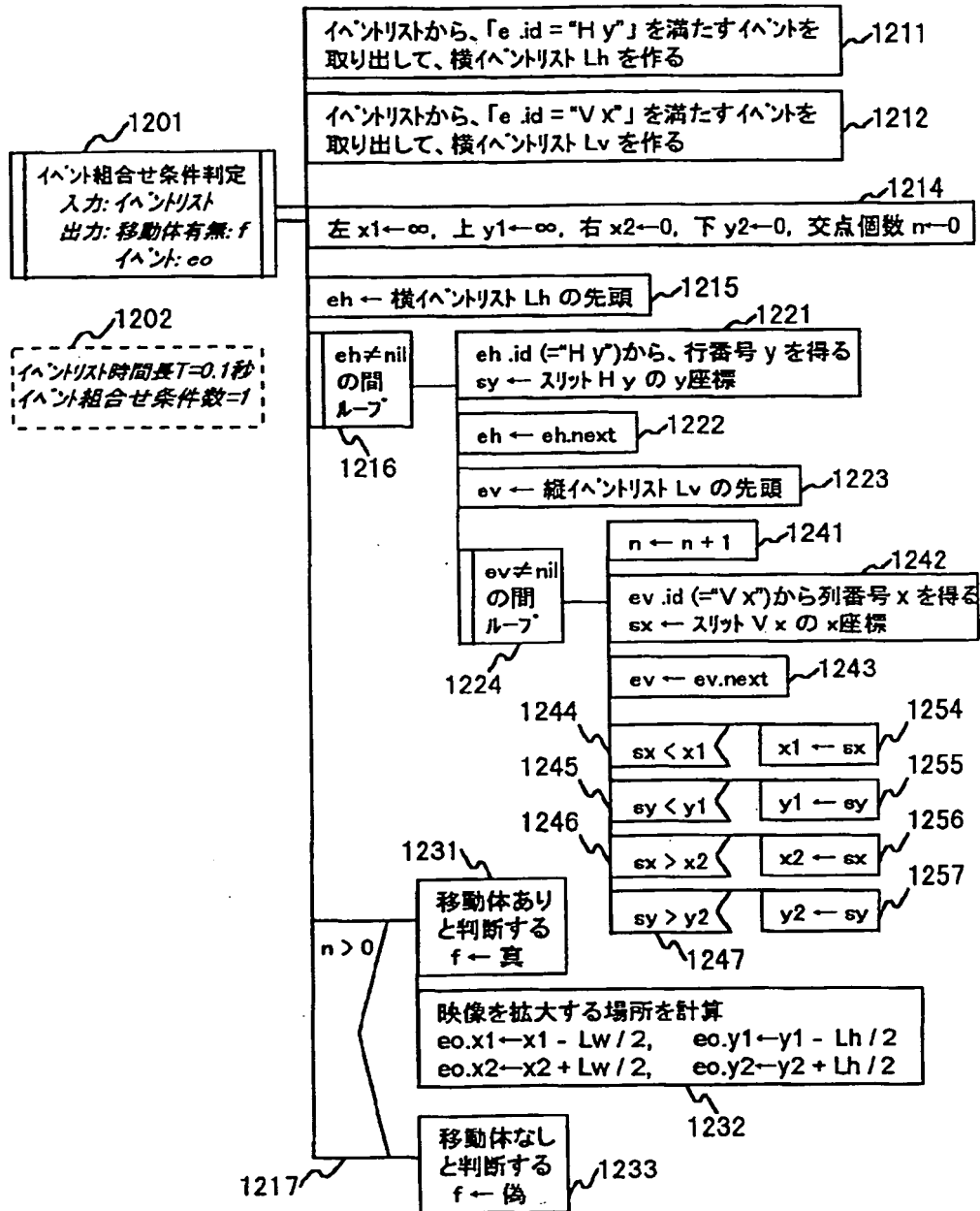
【図 16】

図 16



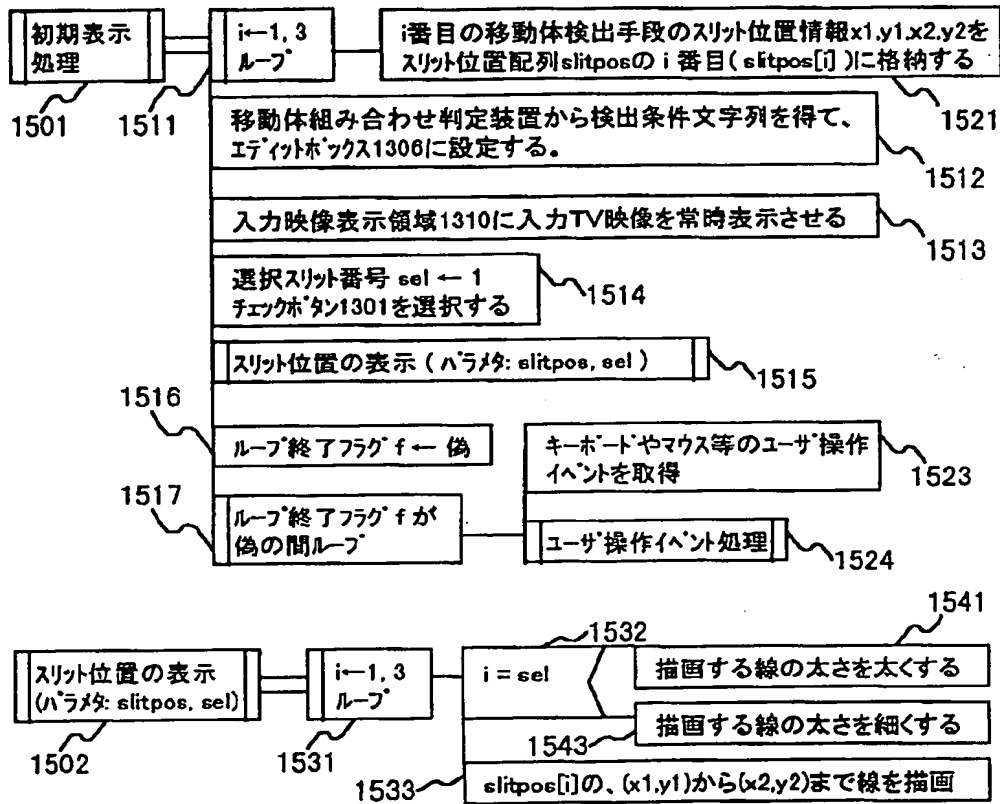
【図 12】

図 12



【図15】

図15



フロントページの続き

(72)発明者 橋本 進二  
秋田県南秋田郡天王町字長沼64 アキタ電  
子株式会社内

(72)発明者 大木 康幸  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア開発本部内

**THIS PAGE BLANK (USPTO)**